

Saimaan ammattikorkeakoulu
Tekniikka, Lappeenranta
Tietotekniikan koulutusohjelma
Tietojärjestelmien kehitys

Kai Kojo

Valokuvausvälinevaraston sähköinen lainausjärjestelmä

Opinnäytetyö 2015

Tiivistelmä

Kai Kojo

Valokuvausvälinevaraston sähköinen lainausjärjestelmä, 50 sivua

Saimaan ammattikorkeakoulu

Tekniikka, Lappeenranta

Tietotekniikan koulutusohjelma

Tietojärjestelmien kehitys

Opinnäytetyö 2015

Ohjaajat: lehtori Martti Ylä-Jussila, Saimaan ammattikorkeakoulu,

pajamestari Heikki Humberg, Saimaan ammattikorkeakoulu

Opinnäytetyön aiheena oli valokuvausvälineiden lainausjärjestelmän toteuttaminen Saimaan ammattikorkeakoulun kulttuurialalle. Työn idea syntyi varaston ylläpitäjän tarpeesta tehostaa lainojen hallinnointia.

Työ toteutettiin web-sovelluksena, joka käyttää hyväkseen MySQL-tietokantaa. Sovelluksen kehittämisvaiheessa käytössä oli XAMPP-ohjelmistopaketti. Ohjelmointikielinä käytettiin PHP- ja JavaScript-ohjelmointikieliä.

Järjestelmällä pystyy hallinnoimaan varastossa olevien laitteiden varauksia ja lainauksia sekä ylläpitämään varaston sisältöä.

Asiasanat: MySQL, PHP, JavaScript, jQuery, XAMPP, tietokanta, web-kehitys

Abstract

Kai Kojo

Lending System for Photographic equipment, 50 Pages

Saimaa University of Applied Sciences

Information technology Lappeenranta

System development

Bachelor's Thesis 2015

Instructors: lecturer, Martti Ylä-Jussila, Saimaa University of Applied Sciences,
shopmaster Heikki Humberg, Saimaa University of Applied Sciences

The subject of the thesis was the implementation of the photographic equipment lending system for Saimaa University of Applied Science's Department of Fine Arts. The idea for this thesis came from the current warehouse caretaker's need to enhance the management of the warehouse and loans.

The work was carried out as a web application that takes advantage of the MySQL database. XAMPP software package was used during the development stage. The programming languages used were PHP and JavaScript.

The system is able to manage bookings and loans of equipment in the warehouse as well as to maintain the contents of the warehouse.

Keywords: MySQL, PHP, JavaScript, jQuery, XAMPP, database, web-development

Sisällys

Termit ja lyhenteet	6
1 Johdanto	8
1.1 Työn taustaa	8
1.2 Tavoitteet	8
2 Ohjelmistotuotannon menetelmät	9
2.1 Perinteiset mallit	9
2.2 Ketterät menetelmät	10
2.3 Ohjelmistotuotannon vaiheet	11
2.3.1 Määrittely	11
2.3.2 Suunnittelu	11
2.3.3 Toteutus	11
2.3.4 Testaus	12
2.3.5 Käyttöönotto	12
2.3.6 Ylläpito	12
2.4 Projektinhallinta	13
2.5 Tuotteenhallinta	13
2.6 Laadunhallinta	13
2.7 Dokumentointi	14
3 Kehitysympäristö	14
3.1 NetBeans	14
3.2 Notepad++	15
3.3 XAMPP	16
3.3.1 Apache	17
3.3.2 MySQL	18
3.3.3 PhpMyadmin	18
4 Käytetyt tekniikat	19
4.1 HTML	19
4.2 PHP	20
4.3 JavaScript	21
4.3.1 jQuery	21
4.3.2 jQuery UI	22
4.4 CSS	22
4.5 SQL	24
4.6 Ajax	25
4.7 JSON	26
4.8 LDAP	26
4.9 Viivakoodi	27
5 Toteutusprojektin kuvaus	27
5.1 Projektinhallinta	27
5.2 Määrittely	28
5.3 Suunnittelu	28
5.4 Ohjelmointi	28
5.5 Testaus	29
5.6 Käyttöönotto	29
5.7 Laadunhallinta	29
6 Lopputuotteen kuvaus	29
6.1 Järjestelmän kuvaus	29
6.2 Laitteisto	30

6.3	Tietokanta	30
6.4	Pääsivu	31
6.5	Kirjautuminen	33
6.6	Varaston selaus	35
6.7	Nimikkeen tarkastelu	36
6.8	Varaus	37
6.9	Lainaus	38
6.10	Palautus	39
6.11	Nimikkeen lisäys	40
6.12	Nimikkeiden hallinta	41
6.13	Varastonhallinta	42
6.14	Raportit	43
6.14.1	Palautusraportti	44
6.14.2	Lainaukset-raportti	44
6.14.3	Inventaarioraportti	45
6.14.4	Varastontilaraportti	46
6.15	Ajastetut tehtävät	46
6.16	Asetukset	47
7	Yhteenveto	47
	Kuvat	48
	Lähteet	49

Termit ja lyhenteet

ACP	Adobe Connect Professional, verkkopalaveriohjelma
AD	Active Directory Hakemistopalvelu ja käyttäjätietokanta
Ajax	Asynchronous JavaScript And XML. Joukko verkkosovelluskehityksen tekniikoita, joilla verkkosovelluksista saadaan vuorovaikutteisempia.
Apache	Avoimeen lähdekoodiin perustuva palvelinohjelmisto.
Avoin lähdekoodi	Tietokoneohjelmien kehitysmenetelmä, jossa käyttäjä voi vapaasti muokata ohjelman koodia tarpeidensa mukaan.
Cron	Ajastuspalvelu Unix-pohjaisille käyttöjärjestelmille.
CSS	Cascading Style Sheets. Tyyliohje, jolla vaikutetaan HTML-sivun ulkoasuun.
DOM	Document Object Model. Ohjelmointirajapinta, jonka avulla voidaan muokata HTML-dokumentin sisältöä.
HTML	HyperText Markup Language. Verkkosivujen tekemisessä käytetty merkintäkieli.
JavaScript	Olioperustainen kieli, jolla voidaan ohjelmallisesti käsitellä verkkosivun objekteja.
LDAP	Lightweight Directory Access Protocol. Hakemistopalveluissa käytetty tunnistusprotokolla
MySQL	My Structure Query Language. Relaatiotietokantaohjelmisto.
Notepad++	Avoimen lähdekoodin tekstieditori.
Palvelin	Palvelin on internetiin liitetty tietokone, jossa on soveltuva ohjelmisto esimerkiksi verkkosivujen säilytystä varten.
PHP	PHP: Hypertext Preprocessor. Palvelin pohjainen ohjelmointikieli, jota käytetään verkkosivujen luonnissa.
PhpMyAdmin	Internet-selaimen kautta käytettävä MySQL-tietokannan hallintatyökalu.

Toimialue	Tietokonelaitteistoverkko, jossa käyttäjätunnuksia, tietokoneita, tulostimia sekä muita laitteita ja niiden käyttöi- keuksia hallitaan palvelinpohjaisesti.
Unix	Laitteistoriippumaton käyttöjärjestelmä. Yleensä myös Linux.
Webfont	Tekniikka, jolla fontti ladataan palvelimelta.
XAMPP	Ilmainen, verkkosivujen kehittämiseen tarkoitettu työka- lu, joka koostuu Apache, MySQL ja PHP ohjelmistoista.

1 Johdanto

Opinnäytetyön kohteena on Saimaan ammattikorkeakoulun kuvataiteen koulutusohjelman valokuvausvälinevaraston sähköisen lainausjärjestelmän kehittäminen. Lainausjärjestelmällä tässä opinnäytetyössä tarkoitetaan ohjelmistoa, jolla voidaan hallita varastoitujen valokuvausvälineiden varausta ja lainausta sekä ylläpitää varastossa olevia valokuvausvälineitä. Työssä käydään läpi myös opinnäytetyön toteutuksessa käytettyjä vapaaseen lähdekoodiin pohjautuvia työkaluja sekä työmenetelmiä.

1.1 Työn taustaa

Saimaan ammattikorkeakoulun kulttuurialan opiskelijoilla on mahdollisuus lainata oppilaitoksen omistuksessa olevia valokuvausvälineitä. Välineet sijaitsevat varastossa ja varaston ylläpitäjän työn helpottamiseksi haluttiin järjestelmä, jolla lainauksia sekä varaston sisältöä voidaan hallita. Aikaisemmin lainaaminen on tapahtunut paperista kirjanpitoa käyttäen, eikä opiskelijoilla ollut käytössään ajankohtaista tietoa mitä valokuvausvälineitä varastossa oli lainattavissa ja niiden yhteensopivuudesta.

1.2 Tavoitteet

Työn tavoitteena on kehittää valokuvausvälineiden lainausjärjestelmä Saimaan ammattikorkeakoulun kulttuurialan käyttöön. Järjestelmällä voidaan pitää kirjaa valokuvausvälineisiin kohdistuneista lainauksista, varauksista ja suorittaa kätevästi lainaus- sekä palautustapahtumat lukemalla välineeseen lisättävä viivakoodi.

Järjestelmä sekä siihen kuuluva tietokanta tulevat sijaitsemaan Saimaan ammattikorkeakoulun palvelimella ja järjestelmään on pääsy vain Saimaan ammattikorkeakoulun opiskelijoilla sekä henkilökunnalla.

2 Ohjelmistotuotannon menetelmät

Tässä luvussa kerrotaan lyhyesti ohjelmistotuotannosta. Ohjelmistojen tuottaminen on vaativa tehtävä. Onnistuneen ohjelmiston tuottamisen varmistamiseksi ohjelmiston kehitystyö jaetaan osiin. Seuraavaan osaan voidaan edetä vasta kun edeltävä osa on saatu tarpeellisilta osiltaan valmiiksi. Ositusta kutsutaan ohjelmistokehityksessä vaihejakomalliksi. (Wikipedia: ohjelmistotuotanto.)

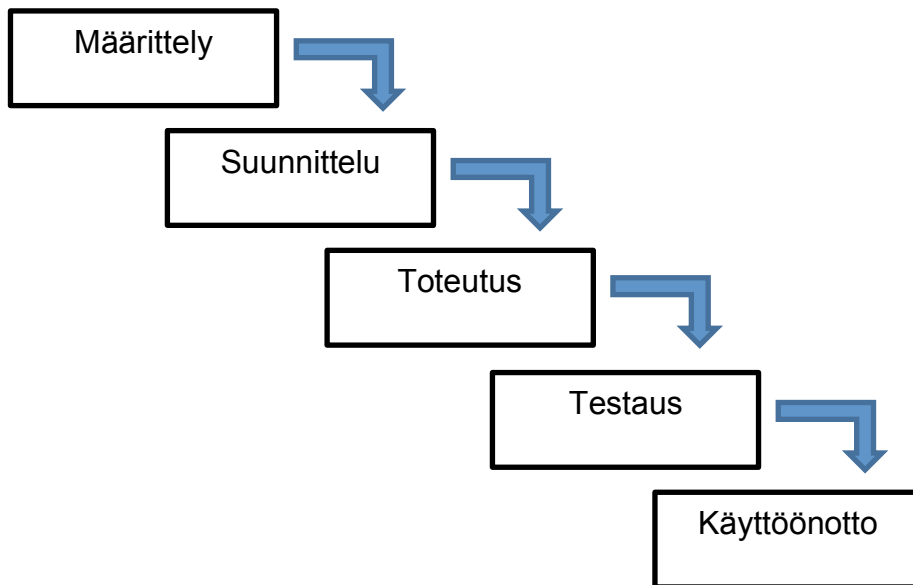
Ohjelmistotuotantoa varten on rajaton määrä erilaisia lähestymistapoja. Eri mallit poikkeavat toisistaan lähinnä siinä, missä kohtaa ja miten eri vaiheita sovelletaan projektin aikana. (Haikala & Mikkonen 2011, s. 29.)

2.1 Perinteiset mallit

Vuosien mittaan on kehitetty useita projektimalleja, joista osa on nimenomaan IT-projekteihin tarkoitettuja. Mallien ympärille on kasvanut merkittävää liiketoimintaa ja niitä räätälöidään eri projekteihin soveltuviksi sekä niistä järjestetään koulutuksia. Moniin projektimalleihin liittyy jopa sertifiointiprosessi, jonka tarkoituksena on varmistaa koulutuksen saaneen henkilön kyky hallita kyseisen mallin mukainen toiminta. (Haikala & Mikkonen 2011, s. 34.)

2.1.1 Vesiputousmalli

Vesiputousmalli on perinteinen vaihejakomalli, jossa vaiheet seuraavat toisiaan tasolta tasolle ja tästä se saa myös nimensä (Kuva 1). Vesiputousmallia käyttäessä tulee vaiheita iteroida myös taaksepäin, jotta siitä saadaan käytännössä toimiva. Mikäli järjestelmän ongelmat havaitaan vasta testausvaiheessa voi kustannusarvio jopa kaksinkertaistua. Usein myös uudempia malleja tarkasteltaessa löytyy niistä usein vesiputousmallin idea. (Haikala & Mikkonen 2011, s. 37.)



Kuva 1. Vesiputousmalli

2.2 Ketterät menetelmät

Ketterillä menetelmillä tarkoitetaan joukkoa ohjelmistotuotantomenetelmiä, joille tyypillistä on kehitettävän ohjelmiston toimivuuden ensisijaisuus, kehitystiimin viestintä ja nopea reagointi muutoksiin. (Wikipedia: Ketterä ohjelmistokehitys.)

Usein ketterät menetelmät jakavat ohjelmistokehityksen lyhyisiin noin viikon tai parin kestäviin pyrähdyksiin. Tämän tarkoitus on minimoida kehitystyön riskejä.

Ketterät menetelmät korostavat ohjelmiston toimivuuden olevan ensisijainen edistymisen mittari, eikä dokumentaatiolla ole samanlaista itseisarvoa kuin perinteisissä ohjelmistotuotannon malleissa. Tällöin siis suunnittelu- ja kehitystyön tapahtuessa samanaikaisesti, suunnitelmiin tulee herkemmin muutoksia. (Wikipedia: Ketterä ohjelmistokehitys.)

Ketteriä menetelmiä on sovellettu tässä opinnäytetyössä viikoittaisilla ohjausryhmän palaverilla sekä vastuun moduulien yksityiskohtaisten suunnittelun toteuttamisesta moduulin toteuttajalle.

2.3 Ohjelmistotuotannon vaiheet

Ohjelmistotuotannon vaihejakomalleissa ohjelmiston kehitys on jaettu määrittelyyn suunnitteluun, toteutukseen, testaukseen sekä käyttöönottoon. Perinteisesti vaiheet seuraavat toisiaan.

2.3.1 Määrittely

Ohjelmistoa määriteltäessä luodaan vaatimusmäärittelydokumentti, jossa kuvataan ohjelmiston vaatimuksia sekä tavoitteita. Siinä määritellään, kuinka ohjelmiston tulisi toimia ja millä keinoilla toiminnallisuudet saavutetaan. Toiminnallisuuksia kuvataan käyttötapausten avulla. Käyttötapaukset puolestaan kuvaavat käyttäjän ja ohjelmiston välistä vuorovaikutusta. Määrittelydokumentti vastaa siis kysymyksiin mitä järjestelmä tekee sekä miten käyttäjä saa järjestelmän tekemään, mitä sen tulee tehdä. Vaatimusmäärittely tehdään yhdessä asiakkaan kanssa. (Wikipedia: Ohjelmistotuotanto.)

2.3.2 Suunnittelu

Suunnittelu on ohjelmistotuotannon vaihe, jossa määritellään ohjelmiston tekninen toteutus. Toisin sanoen ohjelmistosta tai sen osista luodaan tekninen määrittely. Tämä sisältää tiedon siitä kuinka ohjelmisto saadaan toteuttamaan määrittelyvaiheessa määritellyt toiminnot. Tekninen määrittely sisältää muun muassa myös käytetyt ohjelmointikielet, käytettävät ohjelmistokomponentit ja kirjastot sekä komponenttien välinen hierarkia ja rajapinnat. Varsinaista ohjelmakoodia ei vielä tässä vaiheessa luoda. (Wikipedia: Ohjelmistotuotanto.)

Ohjelmat toteutetaan yleensä komponenttien avulla. Yksi komponentti toteuttaa tavallisesti yhden toiminnallisuuden tai joukon yhteen niputettuja ohjelman osia, josta ne helposti löytää. Järjestelmä on tarkoitus jakaa niin pieniin osiin, että yksittäisen komponentin suunnittelu, toteutus ja testaus voidaan yhdistää yhdeksi tehtäväksi. (Haikala & Mikkonen 2011, s 183 – 184.)

2.3.3 Toteutus

Toteutus voi alkaa kun määrittely ja suunnittelu on saatu hyväksytyä. Toteutus käsittää yleensä varsinaisen ohjelmiston koodaamisen suunnitteluvaiheessa

valituilla menetelmillä. Toteutuksen tavoitteena on saada valmiiksi ajettava ohjelmisto, jossa komponentit on integroitu kokonaisuudeksi. Usein ensimmäisen vaiheen toteutuksen jälkeen ohjelmistosta löytyy vielä paljon virheitä, jotka tulisi kuitenkin löytyä testausvaiheessa. (Wikipedia: Ohjelmistotuotanto.)

2.3.4 Testaus

Testaus on ohjelmistokehityksen eräs aikaa vaativimmista osista. Testausvaiheessa ohjelmistosta etsitään ja korjataan ohjelmistossa esiintyvät ohjelmointivirheet eli niin kutsutut bugit. Huolellisesta ohjelmoinnista huolimatta virheitä löytyy käytännössä kaikista ohjelmistoista. Toteutusvaiheessa tarkoitus on saada luotua mahdollisimman virheetön ohjelmisto. Testausvaiheessa puolestaan on tarkoitus löytää mahdollisimman paljon virheitä. Useimmat virheet vaativat korjauksia, jotka puolestaan tarkoittavat lisätyötä ohjelmiston toteuttajalle. Tästä syystä jonkun toisen kuin toteuttajan on testattava ohjelma. Toisena syynä voidaan pitää toteuttajan sokeutta omille virheille. (Haikala & Mikkonen 2011, s.205–209.)

2.3.5 Käyttöönotto

Käyttöönotolla tarkoitetaan ohjelmiston julkaisua asiakkaan käytettäväksi. Se voidaan tehdä, kun ohjelmisto on saatu toimimaan tarpeeksi virheettömästi. Usein käyttöönotto tapahtuu vasta, kun kaikki ohjelmistoa kehittävät tahot ovat siihen tyytyväisiä. Käyttöönotossa ohjelmistosta tehdään kokonaisuus, joka on valmis asennettavaksi. Käyttöönottoon kuuluu myös käyttäjien koulutus, järjestelmään kuuluvan laitteiston asennus sekä tietokantapohjaisissa järjestelmissä usein myös vanhojen tietojen konvertointi. (Wikipedia: Ohjelmistotuotanto.)

2.3.6 Ylläpito

Ylläpito ei ole enää osa ohjelmiston kehitystyötä, vaan sen tarkoitus on pitää ohjelmisto käyttökuntoisena ja raportoida kehitystyön jälkeen löytyvistä toimintavirheistä tuotannolle. Ylläpitoon kuuluu myös uusien versioiden päivitys vanhojen tilalle sekä päivityksestä aiheutuvat muutostoimenpiteet. (Wikipedia: Ohjelmistotuotanto.)

2.4 Projektinhallinta

Projekti on kertaluonteinen tehtävä, jonka tarkoitus on saavuttaa ennalta suunniteltu tavoite. Projekteihin liittyy projektin hallinta, jolla tarkoitetaan resurssien organisointia ja hallintaa niin että projekti voidaan toteuttaa suunnitelman mukaisesti. Projektin resursseja voivat olla esimerkiksi raha, työvoima, raaka-aineet, energia tai käytettävät laitteet ja tilat. Resursseiksi luetaan myös projektin sisäinen viestintä, laatu ja riskit.

Projektinhallinta kuuluu usein projektipäällikön tehtäviin ja hänen tehtävänsä on löytää paras mahdollinen ratkaisu käytettävissä olevien resurssien kesken. Projektipäällikkö kommunikoi myös muiden projektin eri sidosryhmien kanssa, määrittelee ja toteuttaa asiakkaan tarpeet sekä jakaa tehtävät eri työntekijöille. (Wikipedia: Projektinhallinta.)

2.5 Tuotteenhallinta

Tuotteenhallinnan tärkein tehtävä on taata kehittäjille stabiili toimintaympäristö, jossa projektin työntekijät eivät häiritse toistensa työskentelyä tekemällä muutoksia samanaikaisesti samoihin komponentteihin kertomatta muutoksista muille. Tuotteenhallinnan tehtävä on myös arkistoida vanhat versiot, jotta ne on mahdollista saada käyttöön mahdollista korjausta tai testaamista varten. Asiakkaalla saattaa myös syystä tai toisesta olla käytössä tuotteen vanha versio, johon uudemman version komponentit eivät enää ole soveltuvia. Tällöin tuotteesta voidaan tarvittaessa tehdä erillinen versio kyseisen asiakkaan tarpeisiin. Versioista ylläpidettäviä tietoja ovat vähintään versionumero, vastuuhenkilö, muutostentekoajankohta sekä tila. (Haikala & Mikkonen, 2011 s. 172.)

2.6 Laadunhallinta

Testauksen lisäksi onnistuneen ohjelmiston laadun varmistamiseksi käytetään muitakin menetelmiä. Näitä menetelmiä ovat muun muassa tarkastukset ja katselmoinnit, joissa etsitään virheitä jo ohjelmiston suunnitteluvaiheesta lähtien käyttämällä apuna ohjelmistotyön aikana syntyviä dokumentteja. Tarkastusten ja katselmoinnin tehtävänä on myös toimia koulutus- ja tiedotustilaisuuksina, jonka jälkeen kaikilla paikallaolijoilla tulisi olla yhtenäinen näkemys läpikäytävän

materiaalin sisällöstä. Tarkastusta pidetään yleisesti tehokkaampana virheiden etsintäkeinona kuin testausta. (Haikala & Mikkonen, 2011 s. 197-198.)

2.7 Dokumentointi

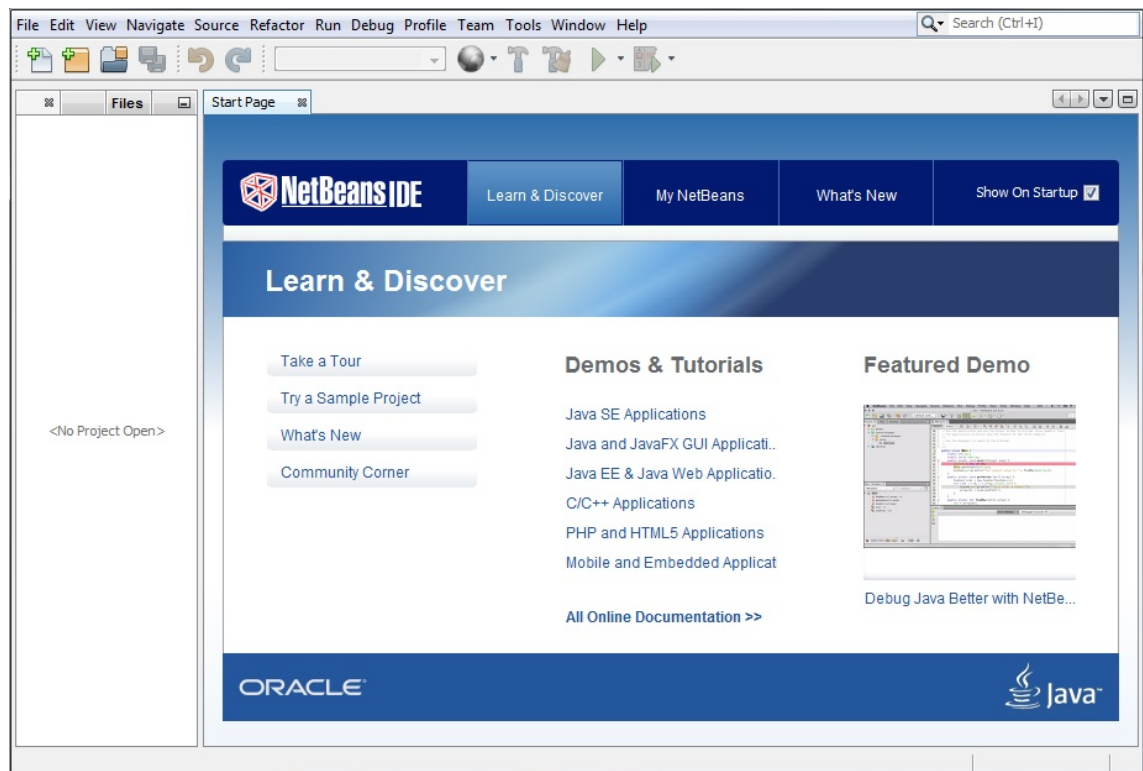
Ohjelmistotyön aikana syntyviä dokumentteja tulee pitää ajan tasalla, jotta niistä on hyötyä projektin edetessä. Dokumentit kannattaa päivittää jo muutosten tullessa, mutta kuitenkin viimeistään projektin päättyessä. Koska dokumentoinnin ylläpito vaatii paljon työtä ei mitään turhaa ole tarpeen dokumentoida. Dokumentteja tehdessä tulee varautua niihin dokumentteihin tuleviin muutoksiin pitämällä kuvaukset mahdollisimman tiiviinä ja näin pyrkiä välttymään dokumentoinnin ylläpidon aikana syntyviltä ristiriitaisuuksilta. (Haikala & Mikkonen, 2011 s. 194-196.)

3 Kehitysympäristö

Tässä luvussa esitellään web-sovelluksen kehityksessä käytetyt kehitystyökalut sekä tietokannan hallintajärjestelmä ja hallintatyökalu. Kehitysvaiheessa sivusto toimii lähiverkossa, eikä se näy tämän ulkopuolelle.

3.1 NetBeans

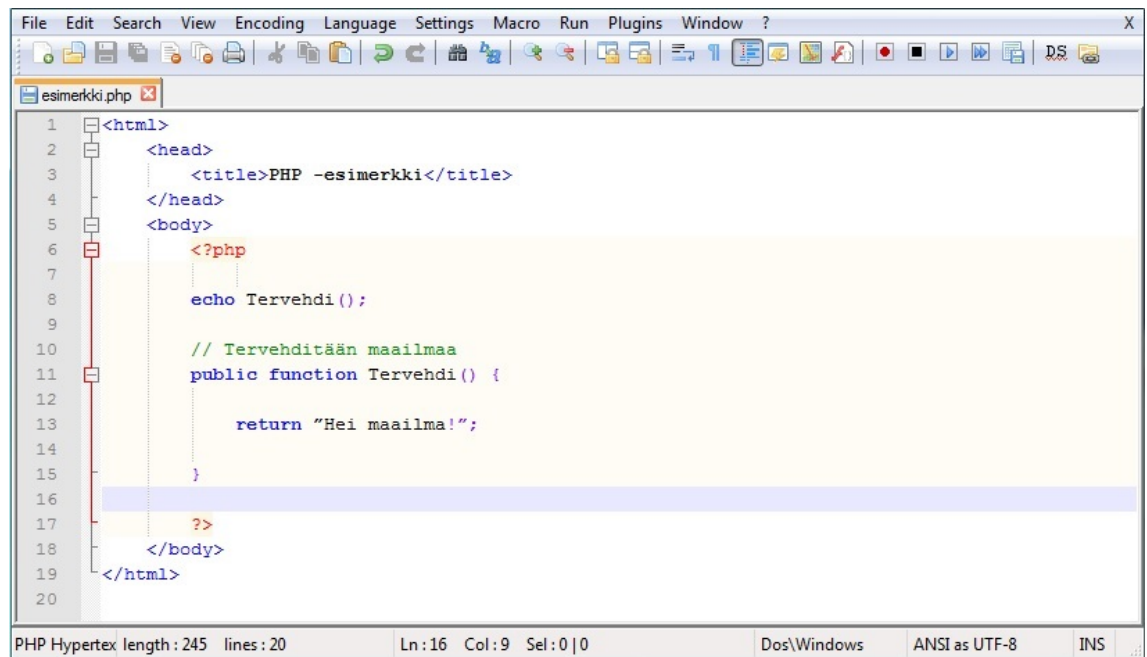
NetBeans on avoimeen lähdekoodiin perustuva ohjelmointiympäristö. (Kuva 2) NetBeans tukee ohjelmistokehitystä Java-, JavaScript-, PHP-, Python-, Ruby-, Groovy-, C-, C++-, Scala- ja Clojure-ohjelmoinikielille ja sitä voi laajentaa kolmannen osapuolen kehittämällä moduuleilla. NetBeansista on saatavana räätälöidyt versiot Java- ja Web -ohjelmien kehitykseen sekä mobiilisovelluksia varten. (Wikipedia: NetBeans.)



Kuva 2. NetBeans

3.2 Notepad++

Notepad++ on Windows-käyttöjärjestelmälle saatava avoimeen lähdekoodiin pohjautuva tekstieditori (Kuva 3). Siinä on tuki useiden eri ohjelmointikielten syntaksin korostusta varten. Notepad++ ei kuitenkaan informoi käyttäjää mahdollisista virheistä koodin syntaksissa, mutta soveltuu nopeiden muokkausten tekemiseen lähdekoodiin sekä koodin tarkasteluun. (Wikipedia: Notepad++.)



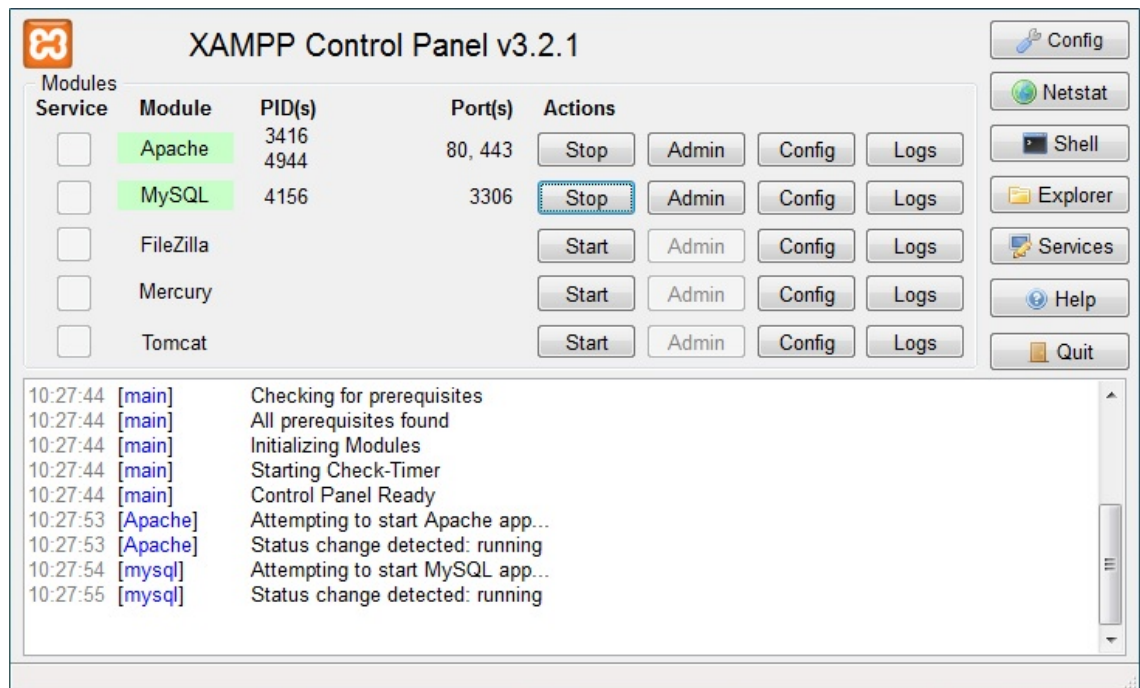
```
1 <html>
2 <head>
3 <title>PHP -esimerkki</title>
4 </head>
5 <body>
6 <?php
7
8     echo Tervehdi();
9
10    // Tervehditään maailmaa
11    public function Tervehdi() {
12
13        return "Hei maailma!";
14    }
15
16    ?>
17
18 </body>
19 </html>
20
```

PHP Hypertext length: 245 lines: 20 Ln: 16 Col: 9 Sel: 0 | 0 Dos\Windows ANSI as UTF-8 INS

Kuva 3. Notepad++

3.3 XAMPP

Web-sivujen kehittämisessä XAMPP on yksi käytetyimmistä työkaluista. Ohjelma perustuu avoimeen lähdekoodiin, ja se on suunniteltu antamaan mahdollisuuden kehittää ja testata kehittämänsä sivuston omalla tietokoneella ilman varsinaista palvelinta. Käytännössä XAMPP:ia voi kuitenkin käyttää verkkosivujen esittämiseen omalta tietokoneelta. Ohjelmistopaketti sisältää muun muassa Apache-palvelinohjelmiston ja MySQL-tietokannan. (Kuva 4). (Wikipedia: XAMPP.)



Kuva 4. XAMPP-kontrollipaneeli

3.3.1 Apache

Apache on usealle käyttöjärjestelmälle saatavissa oleva avoimeen lähdekoodiin perustuva http-palvelinohjelma (Kuva 5). Apachea voidaan laajentaa useilla moduuleilla, joilla palvelin saadaan räätälöityä omia tarpeita vastaavaksi. (Wikipedia: Apache.)



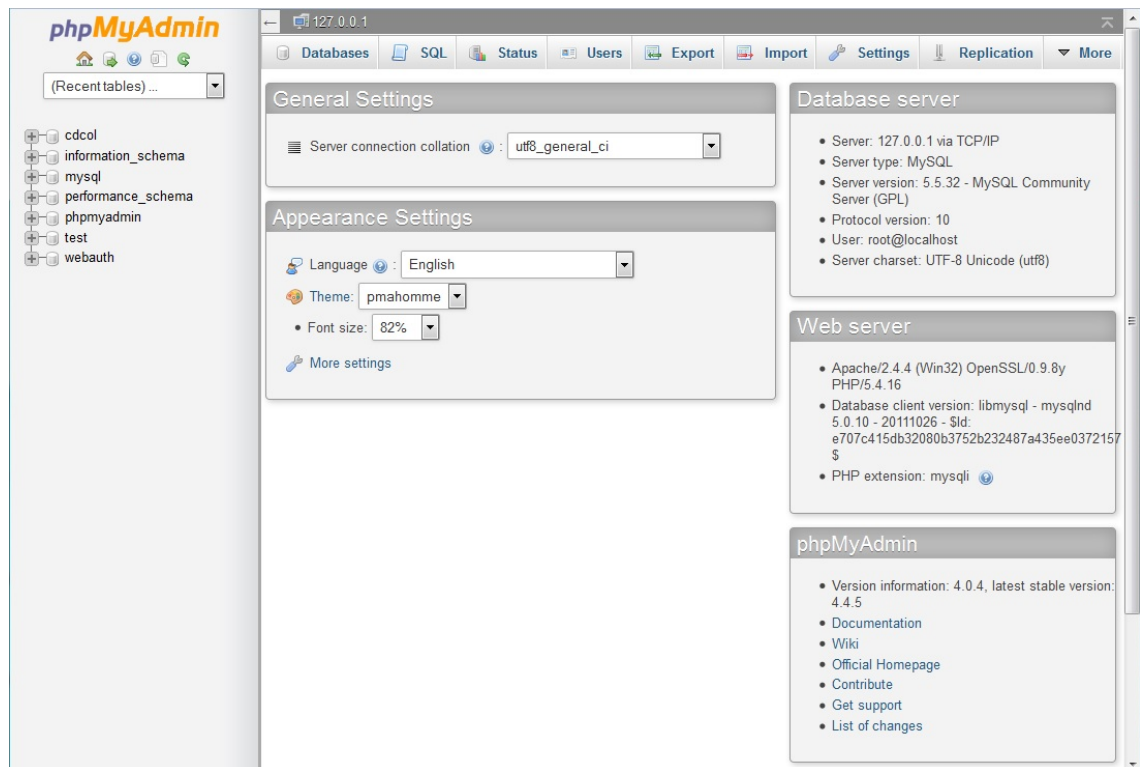
Kuva 5. XAMPP-etusivu palvelimella

3.3.2 MySQL

MySQL on relaatiotietokantaohjelmisto ja se on suosittu web-palveluiden tietokantana. Sitä on mahdollista käyttää usealla ohjelmointikielellä. Useimmin MySQL:n alustana toimii Linux-käyttöjärjestelmällä varustettu Apache-palvelin, jolla myös MySQL:ää käyttävä www-sivusto on julkaistu.

3.3.3 PhpMyadmin

PhpMyAdmin on MySQL-tietokannan hallintatyökalu, jota käytetään www-selaimella. (Kuva 6).



Kuva 6. PhpMyAdmin hallintatyökalu

4 Käytetyt tekniikat

Tässä luvussa käsitellään sivuston toteutuksessa tarvittavia tekniikoita. Sivuston toiminnallisuuden toteutuksessa käytetään HTML-, PHP- sekä JavaScript -ohjelmointikieliä.

4.1 HTML

HTML (Hypertext Markup Language) on avoimesti standardoitu hypertekstin merkitäkieli. HTML tunnetaan kielenä, jota käytetään www-sivujen koodaamisessa. Kieli koostuu sisäkkäisistä ja perättäisistä elementeistä. Koodissa elementtejä edustaa kulmasulkeilla merkityt tunnisteet. Alla on esitelty HTML:n perussyntaksi.

```
<tunniste>elementin sisältö</tunniste>
```

Seuraavana on esimerkki HTML-koodista, joka tulostaa näytölle tekstin ”Hei maailma!”.

```

<html>
  <head>
    <title>HTML-esimerkki</title>
  </head>
  <body>

    <p>Hei maailma!</p>

  </body>
</html>

```

4.2 PHP

PHP on yleinen vapaaseen lähdekoodiin perustuva ohjelmointikieli, joka on käytännössä erikoistettu www-käyttöön ja sitä voidaan käyttää HTML-koodin seassa. PHP:n vahvuuksia on sen samankaltaisuus muiden yleisesti käytössä olevien ohjelmointikielten kanssa sekä suuri määrä erilaisia toimintoja, joita voidaan edelleen lisätä laajennuksilla. PHP-koodi suoritetaan palvelimen puolella ja sillä voidaan generoida koodia, joka lähetetään sivuston lataajalle. (PHP.)

Alla on esimerkki HTML-tunnisteesta, joka sisältää PHP-koodia.

```

<?php // php-koodia ?>

```

Seuraavana on esimerkki PHP-koodista, joka tulostaa näytölle tekstin "Hei maailma!".

```

<html>
  <head>
    <title>PHP-esimerkki</title>
  </head>
  <body>
    <?php

      echo "Hei maailma!";

    ?>
  </body>
</html>

```

4.3 JavaScript

JavaScript on komentosarjakieli, jota käytetään pääasiassa web-ympäristössä. JavaScriptin tärkein sovellus on mahdollisuus lisätä Web-sivuille dynaamista toiminnallisuutta. JavaScriptiä voidaan laajentaa JavaScript-kirjastoilla, joista tässä työssä käytetään jQueryä ja jQuery UI:ta. JavaScriptiä ei tule sekoittaa Javaan, joka on oliopohjainen ohjelmointi kieli. (JavaScript.)

Alla on esimerkki JavaScript-koodista, joka tulostaa näytölle tekstin "Hei Maailma!".

```
<html>
  <head>
    <title>JavaScript-esimerkki</title>
  </head>
  <body>
    <script>

      document.write("Hei maailma!");

    </script>
  </body>
</html>
```

4.3.1 jQuery

jQuery on selaimesta riippumaton avoimen lähdekoodin JavaScript-kirjasto, jonka tehtävä on parantaa ja yksinkertaistaa JavaScriptin ja html :n vuorovaikutusta. jQueryn avulla voidaan tehdä joustavampia ja tehokkaammin toimivia verkkosivuja. jQuery kehitettiin helpottamaan Document Object Model (DOM) -elementtien valintaa, käsittelyä sekä animointia. Toisinaan jQueryä saatetaan käyttää ainoastaan sen mahdollistaman Ajax -sovellusten toteuttamiseen. (jQuery.)

Kirjasto voidaan ottaa käyttöön sisällyttämällä sivun merkkaukseen viittaus haluttuun jQueryn versioon jQuery projektin verkkosivuilla tai lataamalla kirjasto osaksi kehitettävää sivustoa. Tämän jälkeen kirjastoa voidaan käyttää script-lohkojen sisällä.

jQueryn perussyntaksi näyttää seuraavalta:

```
$( valitsin ).toiminto();
```

jQueryn perussyntaksissa "\$" -merkki kertoo rivin olevan jQuery-komento. Seuraavana on valitsin, jolla saadaan haluttu HTML-elementti käsiteltäväksi ja ".toiminto()" on tapahtuma, joka valitulle elementille halutaan tehdä. jQueryssä käytetty valitsin syntaksi perustuu CSS-tyyliohjeissa käytettyihin valitsimiin.

Esimerkki jQuery -komennosta:

```
$( "table tr:not(:first)" ).each( function () { $( this ).toggle(); } );
```

Yllä olevassa esimerkissä näytetty jQuery-komento käy läpi valitsimeksi annetut kaikki sivulta löytyvät HTML-taulukoiden rivit yksitellen lukuun ottamatta taulukoiden ensimmäistä riviä sekä suorittaa niistä jokaiselle funktion, joka piilottaa tai asettaa rivin näkyviin.

4.3.2 jQuery UI

jQuery UI on jQueryä edelleen laajentava vapaan lähdekoodin käyttöliittymälaajennus. Käyttämällä jQuery UI :ta sivustolle saadaan lisättyä sen sisältämiä valmiita efektejä ja hyödyllisiä liitännäisiä. (jQuery UI.)

4.4 CSS

Porrastetulla tyyliohjeella, CSS, (Cascading Style Sheets), tarkoitetaan www-dokumentin ulkoasun määrittelyä. Käytännössä tyyliohjeet koskevat graafista esitystapaa ja erottavat dokumentin ulkoasun dokumentin sisällön rakenteesta. (Weppipankki.)

Alla on esimerkki CSS perussyntaksista.

```
Valitsin { Attribuutti: Arvo; }
```

CSS-kielinen tyyliohje koostuu säännöistä. Käytännössä perussyntaksissa valitsemalla kerrotaan mihin dokumentin elementteihin rivin sisältämien attribuuteille annetut arvot halutaan liittää.

Alla olevassa esimerkissä oleva valitsin, (.menu a), kohdistaa säännön koskemaan kaikkia niitä linkki elementtejä, (a), jotka ovat sellaisen elementin sisällä, mille on annettu luokka-attribuutin, (class), arvoksi "menu". Aaltosulkeiden välissä olevat attribuutit erotellaan toisistaan puolipisteillä. Alla olevan esimerkin sääntö määrittää linkkien värin olevan musta ilman alleviivausta tai muuta koristelua.

```
.menu a { color: #000000; text-decoration: none }
```

Tyyliohje voidaan asettaa www-dokumentin elementeille muutamalla eri tavalla. Yksi vaihtoehto on asettaa tyyli suoraan elementtiin antamalla se suoraan tyyli, (style), attribuuttiin:

```
< h1 style="color: #ffffff" >
```

Toinen vaihtoehto on sisällyttää tyyliohje style-elementtiin HTML -dokumentin head-osioon:

```
<style type="text/css">
    h1 style="color: #ffffff" >
</style>
```

Usein tyyliohjeita tulee sivustolle useita, jolloin ne voidaan sisällyttää erilliseen CSS-tiedostoon, johon viitataan HTML-dokumentin head-osassa. Seuraavassa on Linkkiviittaus CSS-tiedostoon:

```
<link rel="stylesheet" href="tyyli.css" />
```

Samalle elementille voidaan antaa tyyliohje useaan paikkaan. Tällöin tyyliohjeen sijainnilla on merkitystä mikä kilpailevista tyyleistä jää voimaan. Tämä

mahdollistaa www-dokumentin muotoilun muuttamisen dynaamisesti riippuen siitä, missä osassa sivustoa käyttäjä milläkin hetkellä on.

4.5 SQL

SQL on kyselykieli, jolla voidaan käyttää relaatiotietokantoja. Relaatiotietokannoissa tieto esitetään tauluissa, joiden välille luodaan yhteyksiä. Yhteyksien luomiseen käytetään toisen taulun avaintietoa, jota kutsutaan vierasavaimeksi (Foreign Key (FK)). Taulujen yhteyksien luomisessa määritellään myös viite-ehteyksiä, joilla siis määritellään mitä toisen taulun tiedoille tapahtuu jos toisen tietoa muutetaan tai poistetaan. (Wikipedia: SQL.)

Tärkeimpiä SQL:n käyttökäskyjä ovat: SELECT, jolla tietoja voidaan valita. UPDATE, jolla tietoa voidaan päivittää. INSERT, jolla tietoa voidaan lisätä. Ja DELETE, jolla tietoa voidaan poistaa. (Wikipedia: SQL.)

Esimerkkejä SQL-komennoista:

```
SELECT id, tietue FROM taulu WHERE quux = 'xyzy' ORDER BY id DESC;
```

```
UPDATE taulu SET kentta = 'esimerkki' WHERE id = 42;
```

```
INSERT INTO taulu (kentta, toinenkentta) VALUES ('tietoa', 5);
```

```
DELETE FROM taulu WHERE kentta = 123;
```

4.5.1 SQL-injektio

Tietokantapohjaisten sovelluksien kehittämisessä tulee huomioida tahattoman tai tahallisen SQL-injektion mahdollisuus. SQL-injektiolla tarkoitetaan haitallisen SQL-käskyn suorittamista, jota hyökkääjän ei pitäisi pystyä antamaan. SQL-injektoiden torjumiseksi kaikkien käyttäjiltä tulevan tiedon oikeellisuus tulee tarkistaa, niin että SQL:ssä käytetyt erikoismerkit merkitään erikoismerkeiksi oikeassa muodossa. (Wikipedia: SQL-injektio.)

Esimerkki vaarallisesta PHP-koodista:

```
mysql_query("SELECT tietue FROM taulu WHERE id =  
'" . $id . "'");
```

Kun yllä olevaan esimerkin koodiin antaa \$id -muuttujan arvoksi annetaan "; DROP TABLE taulu; --" tietokannalle välittyä alla olevat käskyt, joista jälkimmäinen kääkee poistamaan taulun tietokannasta.

```
SELECT tietue FROM taulu WHERE id = ''; DROP TABLE tau-  
lu; --';
```

4.6 Ajax

Ajax (Asynchronous JavaScript And XML) on web-sovellustekniikka, jonka avulla web-sovelluksista voidaan tehdä vuorovaikutteisempia. Ajaxilla yleensä viitataan asynkronisesti tehtäviin HTTP-pyyntöihin, joilla vaihdetaan pieniä määriä dataa palvelimen ja selaimen välillä. Datan vaihto tapahtuu taustalla niin, ettei koko verkkosivua tarvitse ladata uudelleen muutoksia tehdessä. Vuorovaikutteisuuden lisäksi Ajaxilla pyritään saamaan web-sovelluksista nopeampia sekä kevyempiä. (Wikipedia: AJAX.)

Alla on esimerkki jQuery Ajax -funktion rakenteesta:

```
function Hae(){  
  
    $.ajax( {  
        type: "post",  
        async: false,  
        data: {  
  
            // välitettävät parametrit  
  
        }  
        success: function( data ){  
  
            // palautettavan tiedon käsittely  
  
        }  
    } );  
}
```

4.7 JSON

JSON (JavaScript Object Notation) on avoimen standardin tiedostomuoto. Nimestään huolimatta JSON ei ole riippuvainen JavaScriptistä. JSON:ia käytetään usein tiedonvälitykseen PHP:n ja JavaScriptin välillä Ajaxia käyttäen. (Wikipedia: JSON.)

Seuraavana on esimerkki JSON-tiedostomuodosta:

```
{
  "etuNimi": "Matti",
  "sukuNimi": "Meikäläinen",
  "ikä": 25,
  "osoite": {
    "lähiosoite": "Hämeentie 1",
    "postinumbero": "33100",
    "kaupunki": "Tampere"
  }
}
```

4.8 LDAP

LDAP (Lightweight Directory Access Protocol) on hakemistopalveluille tarkoitettu verkkoprotokolla. Useimmiten LDAP:ia halutaan käyttää käyttäjien tunnistautumiseen ja käyttöoikeuksien tarkistamiseen, niin ettei käyttäjän tarvitse monia eri salasanoja, vaan voidaan käyttää vain yhtä. Käyttäjän tunnistuksessa LDAP-palvelin vastaa, oliko annettu käyttäjätunnus ja salasana oikein ja käyttöoikeuden tarkistuksessa kuuluuko käyttäjä ryhmään, jolla on oikeus käyttää kyseistä resurssia. (Wikipedia: LDAP)

Hyvänä esimerkkinä voidaan pitää usean tietokoneen sisältäviä toimialueita, joissa käyttäjällä on mahdollisuus käyttää mitä tahansa toimialueen konetta omalla tunnuksellaan. LDAP:illa voidaan asettaa kaikkia toimialueen tietokoneita koskevia käyttäjäkohtaisia oikeuksia sekä rajoituksia.

LDAP-palvelin sisältää yleensä käyttäjistä myös muuta tietoa kuin käyttäjätunnuksen ja salasanan. LDAP:ia voidaankin mieltää eräänlaisena henkilötietoluettelona hieman puhelinluettelon tapaan. (Gracion.)

4.9 Viivakoodi

Viivakoodi on tiedon esitystapa, jossa tieto on koodattu optisesti koneella luettavaan muotoon. Käytännössä viivakoodi koostuu tiettyä merkkiä vastaavista mustien ja valkoisten raitojen yhdistelmästä. Viivakoodit on kehitetty morseaakkosista. (Barcoding: Barcode history.)

Viivakoodeista on kehitetty useita muunnelmia. Perinteisesti viivakoodi on lineaarinen, mutta on olemassa myös kaksiulotteisia viivakoodeja. Tässä työssä käytetään viivakoodimuotoa CODE 39 (Kuva 7), joka on käytössä useissa vastaavissa järjestelmissä.



Kuva 7. CODE 39 muodossa oleva viivakoodi (Racoindustries.)

4.9.1 CODE 39

CODE 39 voi sisältää numeroita ja kirjaimia. Jokaista merkkiä viivakoodissa vastaa viisi viivaa ja neljä väliä. Näistä kolme on aina leveitä ja kuusi kapeita. Sarja alkaa ja loppuu määrätyllä aloitus/lopetusmerkillä. Tulostetun viivakoodin pituus riippuu siis koodissa olevien merkkien pituudesta. (Wikipedia: Code 39.)

5 Toteutusprojektin kuvaus

Tässä luvussa käydään läpi lainausjärjestelmän toteutusprojektin vaiheet.

5.1 Projektinhallinta

Projektin hallinnassa käytettiin hyväksi Saimaan ammattikorkeakoululla käytössä olevia välineitä, kuten ACP-verkkopalaveriohjelmaa, jonka välityksellä käytiin suurin osa projektin aikana käydyistä palavereista. Projektin aikana syntynyt

dokumentaatio, lähdekoodi ja muu projektiin liittyvä tallennettiin niin ikään Saimaan ammattikorkeakoulun käytössä olevalle, Microsoft Office 365-ohjelmistopakettiin kuuluvaan OneDrive-tiedostonjakopalveluun. Projektin jakokansioon oli kaikilla projektiin kuuluvilla mahdollisuus päästä. OneDrivestä löytyi myös projektin aikataulu, jota pidettiin yllä koko projektin ajan.

Projektiorganisaatioon kuuluivat seuraavat henkilöt:

Projektipäällikkö	Tomi Kiiveri
Laatupäällikkö	Kai Kojo
Ohjaus	Martti Ylä-Jussila
Asiakkaan edustaja	Heikki Humberg

5.2 Määrittely

Lainausjärjestelmää koskeva määrittely on koottu toiminnallinen määrittelydokumenttiin, josta käy ilmi järjestelmän tavoitteet, käyttöympäristö ja toiminnallisuus. Määrittelydokumenttia pidettiin ajan tasalla projektin aikana siihen tulleiden tarkennusten mukaan. Päävastuu määrittelystä oli projektipäällikkö Tomi Kiiverillä.

5.3 Suunnittelu

Järjestelmän arkkitehtuuri saatiin määrittelyvaiheessa sovitusta toteuttamistavoista. Järjestelmän moduulisuunnittelusta vastasi kunkin moduulin toteuttamisesta vastannut itsenäisesti. Moduulien suunnittelua rajasi määrittelyssä sovitut toteutustavat, tietotarpeet ja ulkoasun kohdalta Saimaan ammattikorkeakoulun muiden sivujen kanssa yhtenäinen ulkoasu.

5.4 Ohjelmointi

Järjestelmä ohjelmoitiin toiminnallisuus kerrallaan aikataulussa sovitun jaottelun mukaisesti. Järjestelmän osien valmistuttua ne integroitiin osaksi järjestelmää. Integroinnista vastasi Kai Kojo.

5.5 Testaus

Testausta varten Kai Kojo teki testaussuunnitelman, joka sisältää testaukseen liittyvät käytännöt sekä testitapaukset. Järjestelmän ohjelmakoodia toteuttaneet suorittivat omaan koodiinsa kohdistuvan yksikkötestauksen. Järjestelmän osien integroinnin aikana suoritettiin integraatiotestausta. Saimaan ammattikorkeakoulun tietotekniikan opiskelijat testasivat järjestelmän.

5.6 Käyttöönotto

Järjestelmän käyttöönotto tapahtui Saimaan ammattikorkeakoulun IT-palvelun avustuksella, joka vastasi järjestelmän asettamisesta palvelimelle sekä tulee jatkossa vastaamaan järjestelmän ylläpidosta. Käyttöönottoon kuului lisäksi järjestelmän pääkäyttäjän koulutus. Tietokannan tietosisältö saatiin varaston ylläpitäjältä, Heikki Humbergilta. Vanhat tiedot konvertoi tietokantaan Kai Kojo. Viivakoodien lisääminen laitteisiin jäi varaston ylläpitäjän tehtäväksi.

5.7 Laadunhallinta

Laadunhallintaa varten tehtiin projektista laadunhallintasuunnitelma, joka sisältää oleelliset projektin tiedonkulkuun ja seurantaan liittyvät asiat. Vastuu laadunhallinnan toteutumisesta oli Kai Kojolla.

6 Lopputuotteen kuvaus

Tämä luku kertoo, kuinka järjestelmä on toteutettu. Luvussa käydään myös läpi järjestelmän toiminta ja käyttötapaukset.

6.1 Järjestelmän kuvaus

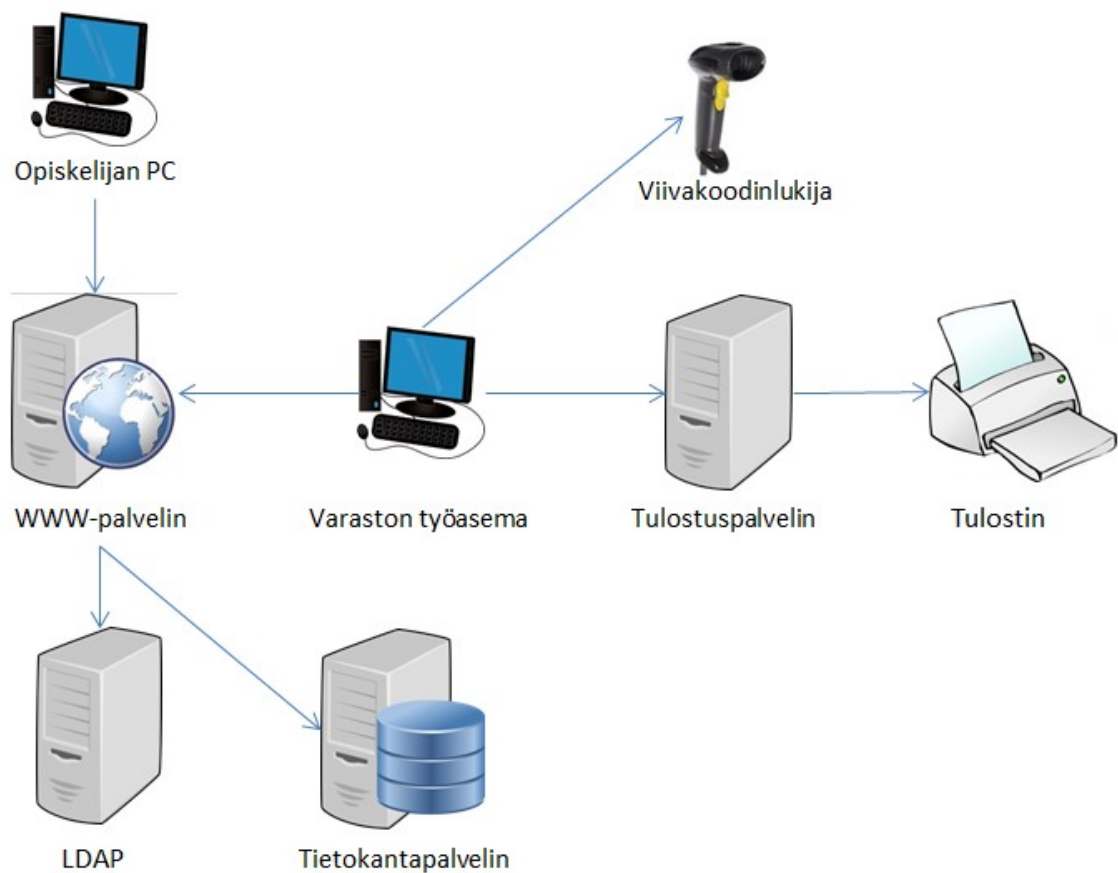
Opinnäytetyönä toteutetulla järjestelmällä on mahdollista ylläpitää varastossa olevia Saimaan ammattikorkeakoulun valokuvausvälineitä sekä seurata niiden lainausta. Järjestelmä on toteutettu tietokantapohjaisena web-sovelluksena.

Järjestelmällä on pääkäyttäjä, joka huolehtii lainausten ja palautusten kirjaamisesta järjestelmään. Pääkäyttäjän on myös ylläpidettävä varaston sisältöä.

Peruskäyttäjät, jotka ovat Saimaan ammattikorkeakoulun opiskelijoita tai henkilökuntaan kuuluvia, voivat järjestelmällä selata ja varata varastossa olevia valokuvausvälineitä.

6.2 Laitteisto

Lainausjärjestelmän laitteisto (Kuva 8) koostuu suurimmaksi osaksi jo Saimaan ammattikorkeakoulun verkossa olevista palvelimista, työasemista sekä tulostimista. Asiakas hankki järjestelmää varten viivakoodinlukijan.

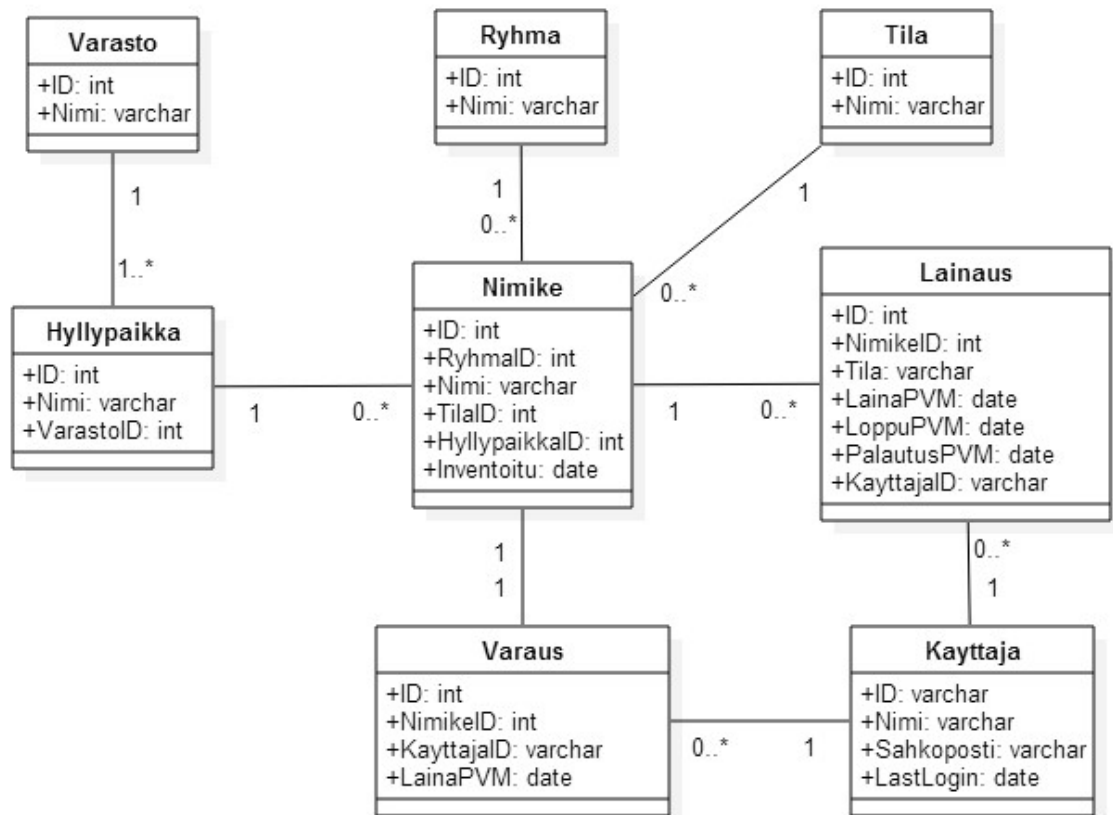


Kuva 8. Järjestelmän laitteet

6.3 Tietokanta

Järjestelmää varten toteutettiin tietokanta, joka sijaitsee järjestelmän käyttöliittymänä toimivan web-sovelluksen kanssa Saimaan ammattikorkeakoulun palvelimella. Tietokannan toteutuksen teki järjestelmän määrittelystä vastannut, pro-

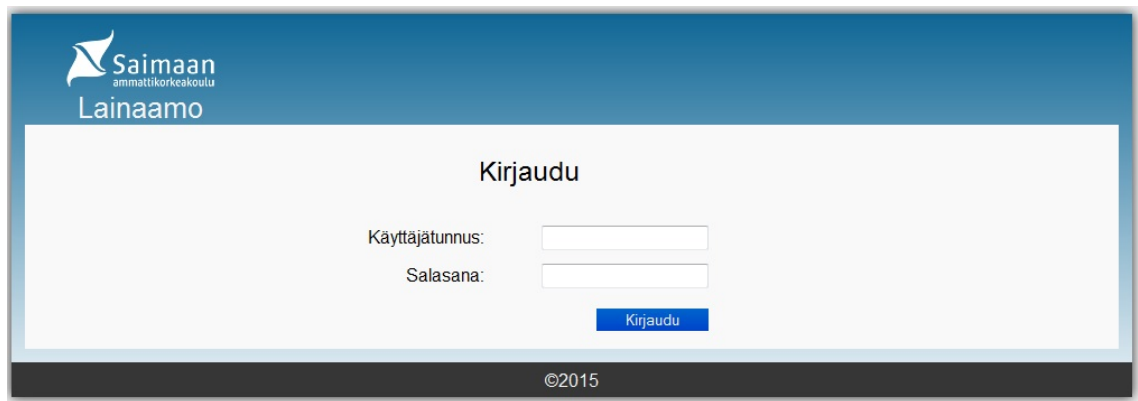
jektipäällikkönä toimiva Tomi Kiiveri. Tietokannan rakenne käy ilmi kuvasta Kuva 9.



Kuva 9. Domain Model

6.4 Pääsivu

Järjestelmää käytetään yhden pääsivun kautta, joka löytyy index.php-tiedostosta. Web-sivuston sisältö on jaettu useaan eri tiedostoon, jotka vastaavat tietyistä toiminnallisuudesta. Haluttu sisältö saadaan selaimen osoiterivillä pääsivulle annetun php:stä löytyvän globaalin \$_GET-muuttujan perusteella. Mikäli Käyttäjä ei ole kirjautunut, eli \$_SESSION-muuttujaa ei ole asetettu, näytetään käyttäjälle kirjautumissivu (Kuva 10).



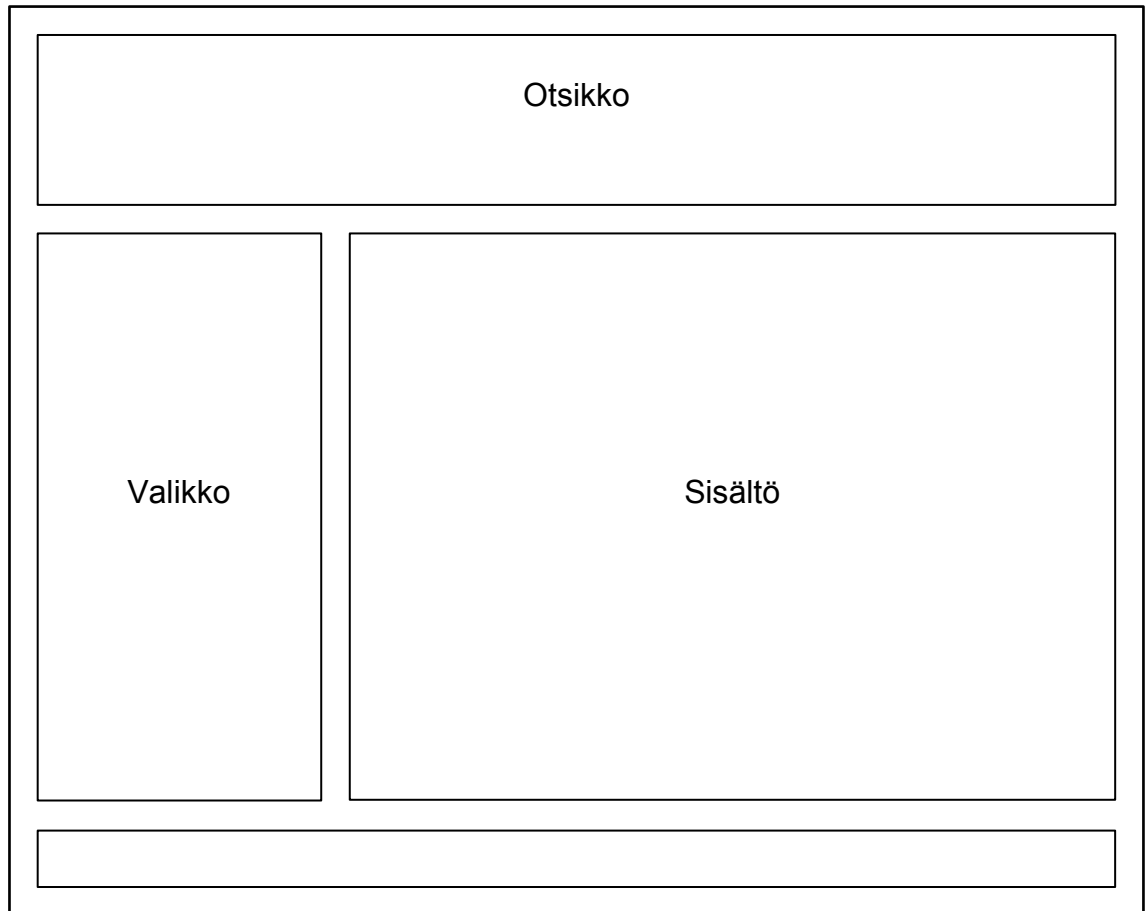
Kuva 10. Sisäänkirjautumisikkuna

Seuraava PHP-koodi valitsee näytettävän sisällön \$_GET-muuttujan avulla.

```
.
.
<?php
    if(!isset($_SESSION["user"])){
        require "content/loginfrm.php";
    }
    else{
        if (isset($_GET["haku"])) {
            require "content/hakufrm.php";
        }
        elseif (isset($_GET["oma"])) {
            require "content/oma.php";
        }
        .
        .
        .
        else{
            require "content/hakufrm.php";
        }
    }
?>
.
.
```

Pääsivu vastaa suuriltaan sivuston asettelusta, joka on toteutettu CSS-tyylitiedoston avulla. Pääsivun tyylitiedostoon on määritetty myös muille sisältöä koskeville sivuille yhteisiä sääntöjä.

Pääsivulla on valikoita, jolla käyttäjä voi siirtyä haluamalleen sisällösivulle. Valikot on toteutettu käyttäen jQuery UI:hin kuuluvaa accordion-komponenttia ja sille on muokattu sivuston tyyliin soveltuva CSS-tyyli.



Kuva 11. Pääsivun rakenne

6.5 Kirjautuminen

Kirjautuminen järjestelmään on toteutettu käyttäen Saimaan ammattikorkeakoululla olemassa olevaa Active Directory -tunnistautumispalvelua. Sama tunnistautumispalvelu mahdollistaa järjestelmän pääkäyttäjän käyttöoikeuksien antamisen. Pääkäyttäjää varten on Saimaan ammattikorkeakoulun Active Directory-verkkoon luotu oma ryhmänsä.

PHP:ssa on mahdollista hakea käyttäjätietoa Active Directorystä LDAP :ia käyttäen. Active Directoryyn otetaan yhteys ldap_bind-komennolla, jolle annetaan parametreiksi ldap_connect-komennolla saatu yhteys, käyttäjänimi sekä sala-

sana. LDAP varmistaa, että käyttäjätunnus ja salasana täsmäävät, jonka jälkeen tietoja voidaan hakea `ldap_search`- sekä `ldap_get_entries` -komennoilla. Tässä tapauksessa tunnistautumisen lisäksi haetaan myös käyttäjän nimi sekä sähköpostiosoite.

Haetuissa tiedoissa on mukana myös ryhmät, joihin käyttäjä kuuluu. Nämä ryhmät käydään läpi ja mikäli käyttäjä kuuluu järjestelmän pääkäyttäjäryhmään, annetaan käyttäjälle pääkäyttäjän oikeudet. Seuraavassa on yksinkertaistettu LDAP-hakuesimerkki.

```
<?php
.
.
$ldap=ldap_connect(LDAP_HOST) or die("Failed to connect");

if ($ldap) {

    $bind=@ldap_bind($ldap,"{$user}@".LDAP_DOMAIN,$psw);

    if($bind){

        $result = ldap_search($ldap, LDAP_DN, $filter, $attr);
        $entries = ldap_get_entries($ldap, $result);

        ldap_unbind($ldap);

    }

}
.
.
.
```

Oikeuksien hallinta tapahtuu PHP:n `$_SESSION`-muuttujalla, joka sisältää käynnissä olevan session muuttujat. Seuraavassa on sessiomuuttujien asetus.

```

<?php
.
.
$_SESSION["user"]=$user;

$_SESSION["name"]=$entries[0]["displayname"][0];
$_SESSION["mail"]=$entries[0]["mail"][0];

foreach($entries[0]['memberof'] as $grps) {

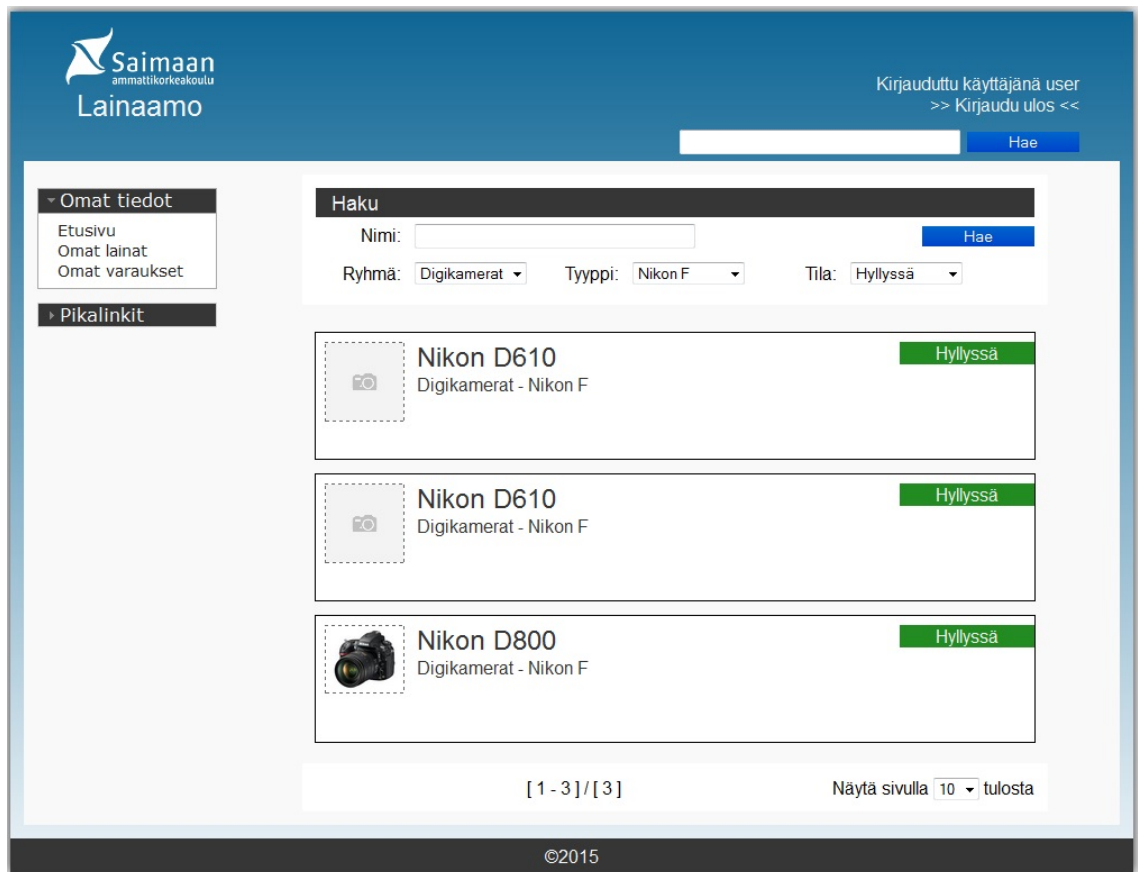
    if (strpos($grps, "=".LDAP_ADMINGRP.",")) {

        $_SESSION["admin"]=1;
        break;
    }
}
.
.

```

6.6 Varaston selaus

Kirjaututtuaan järjestelmään käyttäjälle näytetään varaston selaussivu (Kuva 12). Selauksessa voidaan valita hakuehtoja, joiden mukaan tietokantahaku suoritetaan. Haussa käytetään hyväksi jQuery:n mahdollistamaa Ajax-toimintoa.



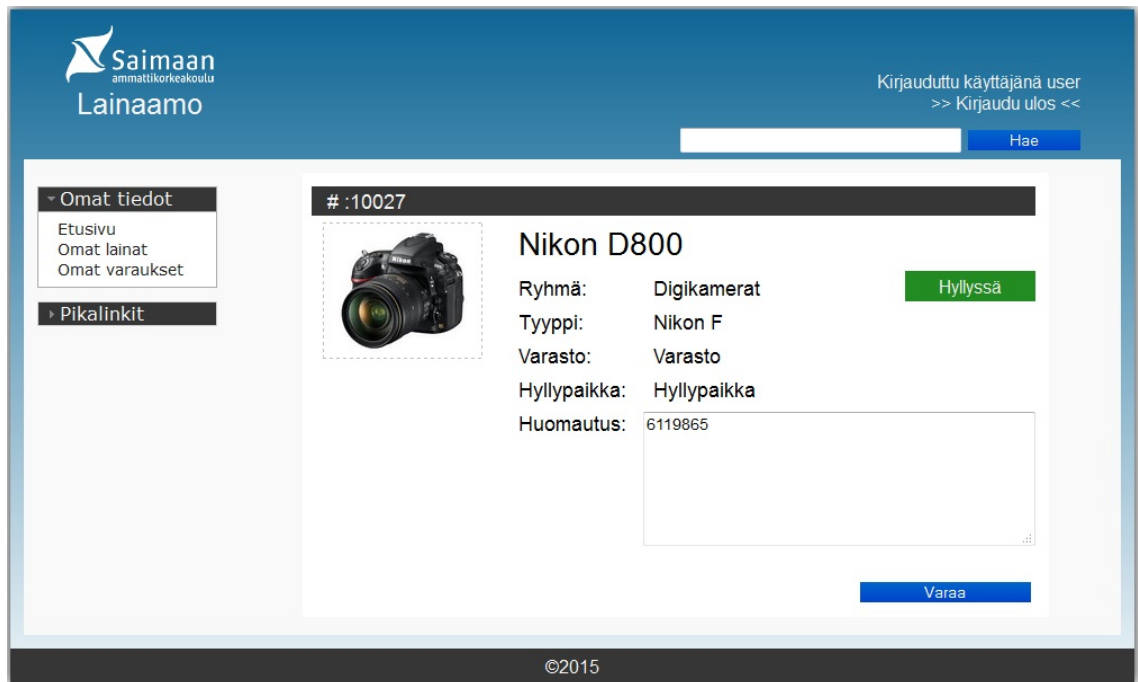
Kuva 12 Varaston selaussivu

6.7 Nimikkeen tarkastelu

Nimikkeen tarkastelusivulla näkyvät nimikkeen yksityiskohtaisemmat tiedot. Näitä ovat nimikkeen tunnus, nimi, ryhmä, tyyppi, varasto, hyllypaikka sekä huolto.

Tarkastelussa näkyy myös, onko kyseinen nimike hyllyssä, lainassa vai onko se varattu. Varatuista ja lainassa olevista välineistä käy ilmi myös varausten lukumäärä. Sivulla on myös varauspainike, jota painamalla varauksen pääsee tekemään.

Pääkäyttäjää varten sivulla on myös painikkeet nimikekohtaisen viivakoodin tulostamiselle sekä nimikkeen muokkaamiselle.

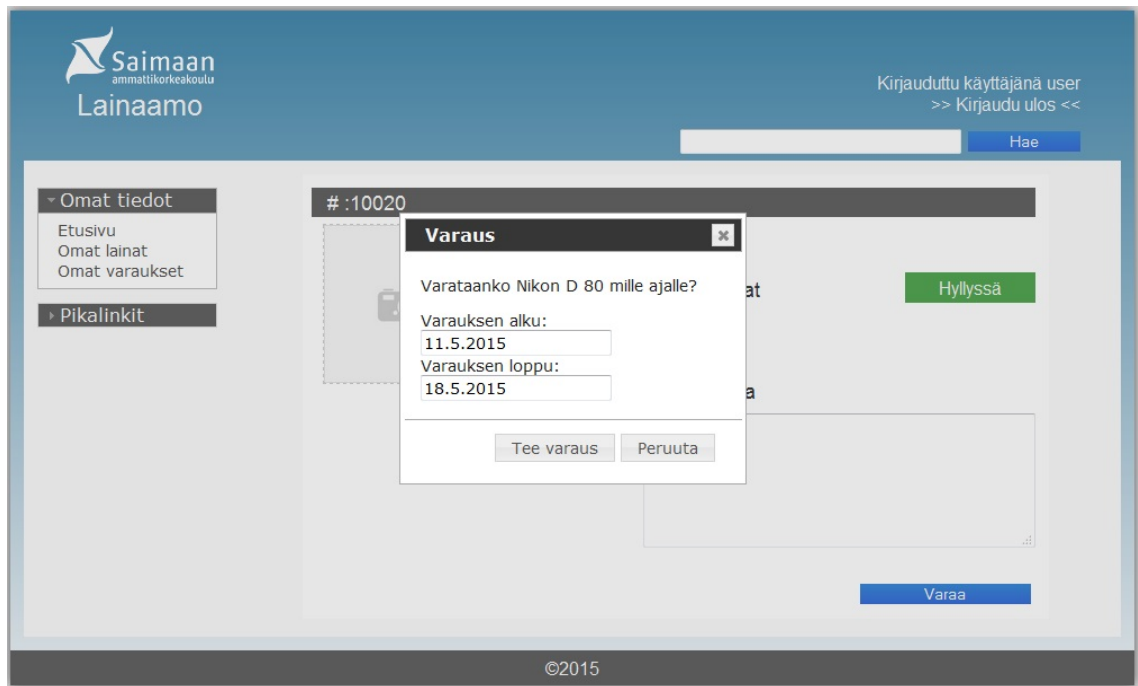


Kuva 13 Nimikkeen tarkastelusivu

6.8 Varaus

Peruskäyttäjällä on varaston selauksen lisäksi mahdollisuus myös varata varastossa olevia valokuvausvälineitä. Varauksen pääsee tekemään nimikkeen omalta yksityiskohtaisemmalta näyttösivulta (Kuva 13). Tälle sivulle pääsee varaston selauksen kautta.

Varaus on toteutettu jQuery UI:stä löytyvillä dialog- sekä datepicker-komponenteilla. Käyttäjän painettua Varaa-painiketta avautuu siis dialogi, josta varauksen alkua ja loppukohdan saa valittua (Kuva 14). Dialog- ja datepicker-komponenteille on valikossa käytetyn accordion-komponentin tavoin tehty muokattu versio jQuery UI:n mukana tulevasta CSS-tyylistä.




Kuva 14 Varaus-ikkuna

6.9 Lainaust

Lainauksen pääsee tekemään lainaustoimintoa varten olevalta sivulta (Kuva 15), jonne pääsee pääkäyttäjän valikosta. Kyseinen valikko näkyy siis ainoastaan pääkäyttäjälle.

Lainausta tehtäessä järjestelmään syötetään lainaajan tunnus, jota käytetään myös järjestelmään kirjautumisessa, käytännössä siis opiskelijanumero. Tämän jälkeen voidaan lukea lainattavista välineistä niihin kiinnitetty viivakoodi ja sen tiedot haetaan lainattavien välineiden listaan. Lopuksi vielä asetetaan päivämäärä, jolloin lainatut välineet tulee viimeistään palauttaa. Lainaustapahtuma tallentuu järjestelmään Tallenna-painikkeesta.



Saimaan
ammattikorkeakoulu

Lainaamo

Kirjautunut käyttäjänä admin
>> Kirjaudu ulos <<

Omat tiedot

Etusivu
Omat lainat
Omat varaukset

Ylläpitovalikko

Lainaamo
Palautus
Varaukset
Uusi nimike
Varastonhallinta
Raportit

Pikalinkit

Lainaamo

Lainaaaja:

Nimi: user

Sähköposti: user@no.mail

Koodi:

Nimike

10020 - Nikon D 80	<input checked="" type="checkbox"/>
10120 - Nikon AF-S DX ED 18-70/3.5-4.5	<input checked="" type="checkbox"/>

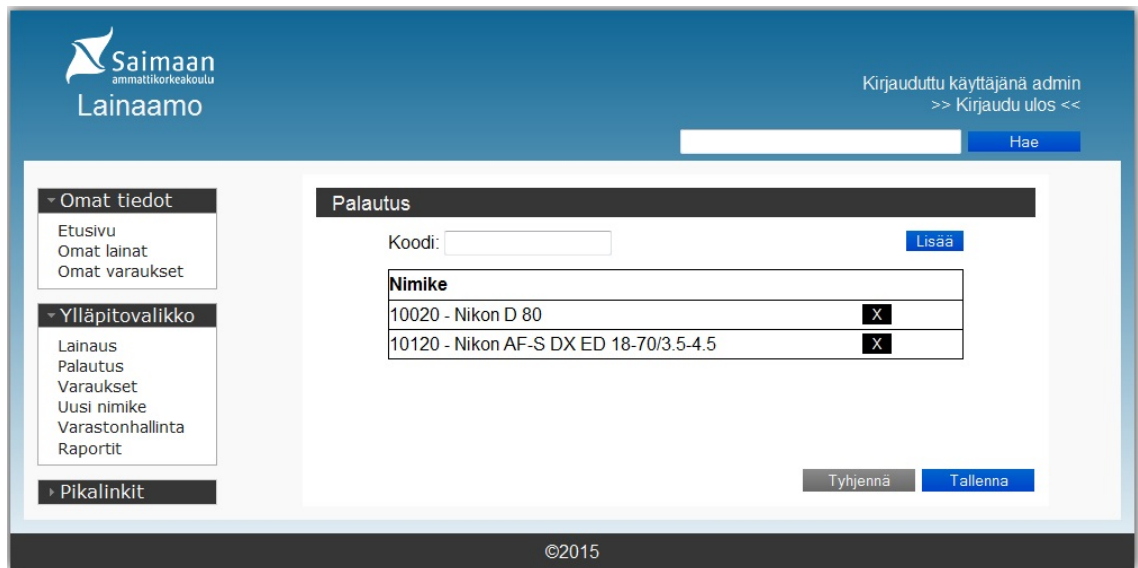
Palautuspäivä:

©2015

Kuva 15 Lainaamo-sivu

6.10 Palautus

Valokuvausvälineiden palautusta varten on oma sivunsa (Kuva 16). Sivun toiminta on samankaltainen kuin lainauksessa. Palauttajaa ei kuitenkaan tarvitse erikseen ilmoittaa, vaan palautustapahtuma päivittää tietokannan tauluista valokuvausvälineen palautetuksi.



Kuva 16 Palautus-sivu

6.11 Nimikkeen lisäys

Uuden nimikkeen lisäys varastoon onnistuu sille tarkoitetulta sivulta (Kuva 17), jonne pääsee ylläpitovalikon Uusi nimike -linkistä. Nimike voidaan lisätä, kun sen tietoihin on asetettu halutut arvot pudotuslistoista ja tekstikentistä.

Nimikkeelle on mahdollista lisätä myös kuva sekä liitetiedosto, jotka tallennetaan palvelimelle Nimikkeen tallennuksen yhteydessä. Kuvien sekä liitteiden sallitut tiedostomuodot voi määritellä asetustiedostossa.

Saimaan ammattikorkeakoulu Lainaamo

Kirjautunut käyttäjänä admin
>> Kirjaudu ulos <<

Hae


Omat tiedot
Etusivu
Omat lainat
Omat varaukset

Ylläpitovalikko
Lainaus
Palautus
Varaukset
Uusi nimike
Varastonhallinta
Raportit

Pikalinkit

:10206

Uusi


Lähetä kuva

Ryhmä:
Tila:
Typpi:
Varasto:
Hyllypaikka:
Huomautus:

Liitteet: [Lisää tiedosto](#)

Peruuta Tallenna

©2015

Kuva 17 Nimikkeen lisäyssivu

6.12 Nimikkeiden hallinta

Nimikkeiden hallinta onnistuu pääkäyttäjänä, Nimikkeen tarkastelu -sivulta löytyvää Muokkaa-painiketta painamalla (Kuva 18). Nimikkeen muokkaus perustuu täysin samaan periaatteeseen kuin nimikkeen lisäys. Muokkauksessa vain päivitetään nimikkeen tiedot uusilla.

Kirjautunut käyttäjänä admin
>> Kirjautu ulos <<

Omat tiedot

Etusivu
Omat lainat
Omat varaukset

Ylläpitovalikko

Lainaus
Palautus
Varaukset
Uusi nimike
Varastonhallinta
Raportit

Pikalinkit

:10027

Tulosta viivakoodi

Nikon D800

Ryhmä:

Digikamerat

Digikamerat

Hyllyssä

Hyllyssä

Tyyppi:

Nikon F

Nikon F

Varasto:

Varasto

Varasto

Hyllypaikka:

Hyllypaikka

Hyllypaikka

Huomautus:

6119865

Liitteet:

Lisää tiedosto

©2015

Kuva 18 Nimikkeen hallintänäkymä

6.13 Varastonhallinta

Varaston hallinta on jaettu kahteen osaan, joista toisessa voidaan hallita varastoja sekä niissä olevia hyllypaikkoja (Kuva 19) ja toisessa nimikkeille annettavia ryhmiä sekä tyyppejä (Kuva 20). Osien jaottelu on tehty jQuery UI:n tabs-komponentilla kahdelle välilehdelle.

Valittaessa varastonhallinnasta hyllypaikka, ryhmä tai tyyppi listaa sivu kuhunkin niistä liittyvät nimikkeet, joita pääsee tarkemmin tarkastelemaan painamalla nimikkeen nimenä olevaa linkkiä.

Saimaan ammattikorkeakoulu
Lainaamo

Kirjaututtu käyttäjänä admin
>> Kirjaudu ulos <<

Hae

Omat tiedot
Etusivu
Omat lainat
Omat varaukset

Ylläpitovalikko
Lainaus
Palautus
Varaukset
Uusi nimike
Varastonhallinta
Raportit

Pikalinkit

Varastot Ryhmät

Varasto: Varasto Varasto Muokkaa

Hyllypaikka: Uusi Lisää

Hyllypaikka	Sisältö
10000	- Gossen Variosix F 2
10001	- Gossen sixtomat Flash
10002	- Gossen sixomat digital
10003	- Gossen starlite
10004	- Variosix F2
10005	- Gossen (pieni)
10006	- PFM prolinca salamamittari
10007	- Gossen digiflash
10008	- Gossen digifidus
10009	- Canon 20 D
10010	- Canon 60 D

Kuva 19 Varastonhallintasivu – Varastot

Saimaan ammattikorkeakoulu
Lainaamo

Kirjaututtu käyttäjänä admin
>> Kirjaudu ulos <<

Hae

Omat tiedot
Etusivu
Omat lainat
Omat varaukset

Ylläpitovalikko
Lainaus
Palautus
Varaukset
Uusi nimike
Varastonhallinta
Raportit

Pikalinkit

Varastot Ryhmät

Ryhmä: Uusi Taustakangas Lisää

Tyyppi: Nikon F Nikon F Muokkaa Poista

Nimike
10026 - Nikon D800
10027 - Nikon D800
10028 - Nikon D610
10029 - Nikon D610
10065 - Nikon FM
10066 - Nikon FM-2
10067 - Nikon FM
10068 - Nikon F5

Kuva 20 Varastonhallintasivu – Ryhmät

6.14 Raportit

Raporttien luomista ja tulostusta varten on olemassa Raportit-sivu. Sivun on jaettu neljään välilehteen, joista kullakin voi määritellä halutun raportin asetuksia.

Raportit-sivulla on myös mahdollisuus tulostaa viivakoodit yksitellen tai kaikki kerralla.

Viivakoodien tulostusta varten on sivulla tarvittaessa ladattavissa fontti viivakoodia varten. Useimmissa tapauksissa viivakoodit tulostuvat kuitenkin webfontin ja CSS-tyylissä määritellyn @font-face-säännön ansiosta. Webfont-ominaisuudella voidaan käyttää palvelimella olevaa fonttia osana web-sivuja.

@font-face-säännön määrittely tapahtuu seuraavasti:

```
@font-face {  
    font-family:"3 of 9 Barcode";  
  
    src:url("3OF9.eot?") format("eot"),  
        url("3OF9.woff") format("woff"),  
        url("3OF9.ttf") format("truetype"),  
        url("3OF9.svg#3Of9Barcode") format("svg");  
  
    font-weight:normal;  
    font-style:normal;  
}
```

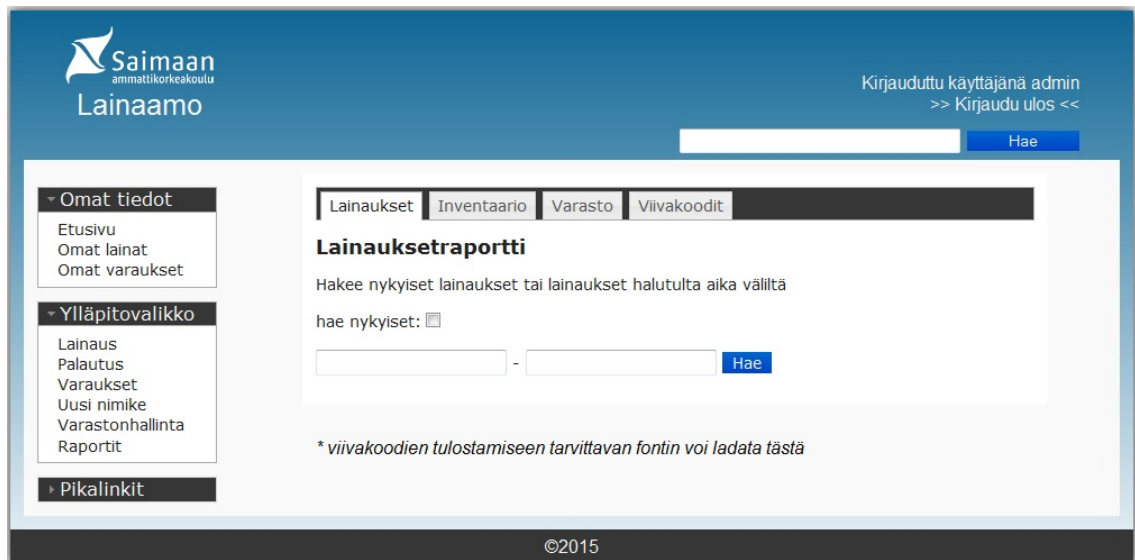
Raporteista Palautusraportti aukeaa automaattisesti palautuksen yhteydessä, eikä sitä voi avata Raportit-sivulta.

6.14.1 Palautusraportti

Palautusraportti aukeaa, kun pääkäyttäjä on tehnyt palautustapahtuman. Raportista käy ilmi palautettujen valokuvausvälineiden nimi sekä hyllypaikka, joka on välinettä varten varattu. Raportin tarkoitus on helpottaa ja nopeuttaa palautettujen välineiden sijoittamista varastoon palautuksen jälkeen.

6.14.2 Lainaukset-raportti

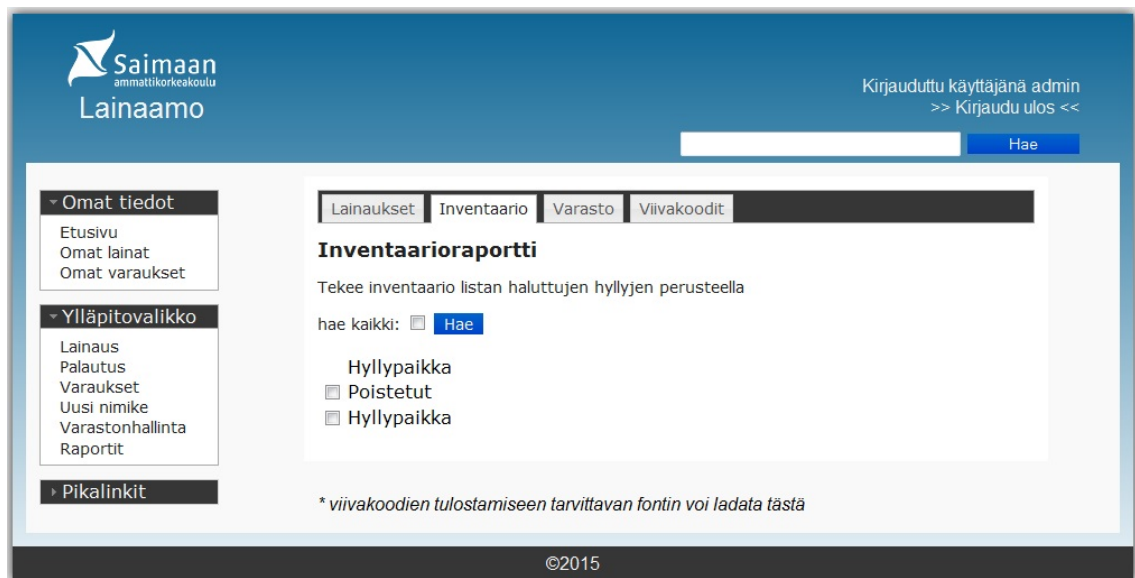
Lainaukset-raportin tehtävä on helpottaa lainausten seuraamista. Raportin luomisivulta (Kuva 21) saadaan valita, halutaanko raportti vain nykyisistä, meneillään olevista lainauksista vai kaikki lainaukset tietyltä ajanjaksolta.



Kuva 21 Lainaukset-raportin luontisivu

6.14.3 Inventaarioraportti

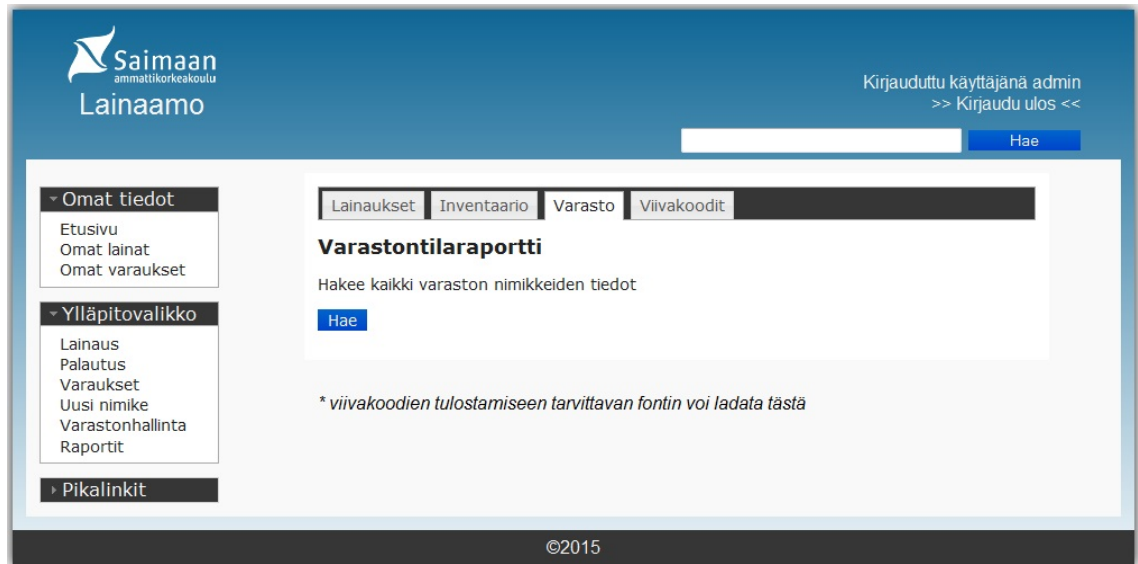
Inventaarioraportin tarkoituksena on toimia apuna varaston inventoinnissa. Se voidaan luoda hyllypaikkakohtaisesti tai sisältää kaikki varaston nimikkeet yhteen raporttiin.



Kuva 22 Inventaarioraportin luontisivu

6.14.4 Varastontilaraportti

Varastontilaraportti listaa kaikki järjestelmässä olevat nimikkeet ja niiden oleellimmat tiedot (Kuva 23).



Kuva 23 Varastontilaraportin luontisivu

6.15 Ajastetut tehtävät

Järjestelmään kuuluu laitteiston, tietokannan ja käyttöliittymänä toimivan web-sivun lisäksi myös muutamia ajastettuja tehtäviä. Ajastetut tehtävät ajetaan päivittäin Cron-ajastuspalvelun avulla. Tehtävät on koottu yhteen PHP-tiedostoon.

Tehtäviin kuuluu automaattinen sähköpostilla lähetettävä huomautus myöhässä oleville palautuksille. Huomautuksia lähetetään kahdesti, joista jälkimmäisellä kerralla ilmoitetaan myöhästymisestä myös järjestelmän pääkäyttäjälle, jonka jälkeen pääkäyttäjä voi ottaa yhteyttä lainaajaan.

Toinen automaattisesti tehtävä työ on tietokannan siivous käyttämättömistä varauksista sekä järjestelmää pitkään käyttämättä olleiden käyttäjien ja heidän lainaushistorian poistaminen.

6.16 Asetukset

Järjestelmän ylläpidon helpottamiseksi järjestelmän web-sivustoa varten on luotu asetustiedosto, `lainaamo_config.php`, jonka sisältöä muokkaamalla voidaan vaikuttaa joihinkin järjestelmän ominaisuuksiin. Tiedostoon on kerätty muuttujia, joita odotetaan tarvittavan muuttaa järjestelmän elinkaaren aikana. Muuttujat on tässä järjestelmässä toteutettu vakioina sekä taulukoina riippuen siitä, muuttaako asetus yhtä vai useampaa arvoa.

7 Yhteenveto

Opinnäytetyön tavoitteena oli kehittää Saimaan ammattikorkeakoulun kulttuurialan käyttöön valokuvausvälineiden lainausjärjestelmä, jolla voi pitää kirjaa varastossa olevista valokuvausvälineistä sekä niiden lainauksista ja varauksista. Tässä mielestäni onnistuttiin hyvin.

Lainausjärjestelmän toteuttaminen web-pohjaisena sovelluksena opinnäytetyönä oli mielenkiintoinen projekti. Aikaisempi kokemus verkkosivujen toteuttamisesta oli lähinnä ollut yksinkertaisia verkkosivuja. Suurena haasteena pidin projektin nopeaa aikataulua, josta johtuen valmiin järjestelmän lähdekoodissa olisi varmasti vielä paljon, mitä voisi tehostaa.

Työssä pääsi soveltamaan opittuja internetohjelmointitaitoja MySQL- ja PHP -ohjelmointikielillä sekä paremmin tutustumaan JavaScript- ja etenkin jQuery-skriptaukseen.

Pieniä tietokantopohjaisia web-sovelluksia on vielä mahdollista tehdä alusta alkaen itse. Markkinoilla on kuitenkin web-sovellusten kehitykseen useita ohjelmistokehys- ja sisällönhallintajärjestelmiä, joilla suurempien ja monimutkaisempien sovellusten toteuttaminen helpottuu huomattavasti.

Kuvat

Kuva 1. Vesiputousmalli, s. 10

Kuva 2. NetBeans, s. 15

Kuva 3. Notepad++, s. 16

Kuva 4. XAMPP-kontrollipaneeli, s. 17

Kuva 5. XAMPP-etusivu palvelimella, s. 18

Kuva 6. PhpMyAdmin hallintatyökalu, s. 19

Kuva 7. CODE 39 muodossa oleva viivakoodi, s. 27

Kuva 8. Järjestelmän laitteet, s. 30

Kuva 9. Domain Model, s. 31

Kuva 10. Sisäänkirjautumisikkuna, s. 32

Kuva 11. Pääsivun rakenne, s. 33

Kuva 12 Varaston selaussivu, s. 36

Kuva 13 Nimikkeen tarkastelusivu, s. 37

Kuva 14 Varaus-ikkuna, s. 38

Kuva 15 Lainaus-sivu, s. 39

Kuva 16 Palautus-sivu, s. 40

Kuva 17 Nimikkeen lisäyssivu, s. 41

Kuva 18 Nimikkeen hallintanäkymä, s. 42

Kuva 19 Varastohallintasivu – Varastot, s. 43

Kuva 20 Varastohallintasivu – Ryhmät, s. 43

Kuva 21 Lainaukset-raportin luontisivu, s. 45

Kuva 22 Inventaarioraportin luontisivu, s. 45

Kuva 23 Varastontilaraportin luontisivu, s. 46

Lähteet

Barcoding: Barcode history.

http://www.barcoding.com/information/barcode_history.shtml. Luettu 11.5.2015

Gracion. What is LDAP. <http://www.gracion.com/server/whatldap.html>. Luettu 18.5.2015

Haikala, I & Mikkonen, T. 2011. Ohjelmistotuotannon käytännöt. Helsinki: Talentum

JavaScript. W3Schools. http://www.w3schools.com/js/js_intro.asp . Luettu 7.5.2015

jQuery. The jQuery Project. <http://jquery.com/>. Luettu 7.5.2015

jQuery UI. jQuery UI Team. <http://jqueryui.com/>. Luettu 7.5.2015

PHP. The PHP Group. <http://www.php.net/manual/en/intro-what-is.php>. Luettu 7.5.2015

Racoindustries: Barcode Generator.

<http://www.racoindustries.com/barcodegenerator/1d/code-39.aspx>. Luettu 19.5.2015

Weppipankki. Johdanto CSS-tyyliehdotuksien käyttöön Web-sivuilla.

<http://weppipankki.com/css/tekstit/cssintro.htm>. Luettu 7.5.1015

Wikipedia: AJAX. [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)). Luettu 10.5.2015

Wikipedia: Apache. [https://fi.wikipedia.org/wiki/Apache_\(palvelinohjelma\)](https://fi.wikipedia.org/wiki/Apache_(palvelinohjelma)). Luettu 21.5.2015

Wikipedia: Code 39. http://en.wikipedia.org/wiki/Code_39. Luettu 15.5.2015

Wikipedia: JSON. <https://en.wikipedia.org/wiki/JSON>. Luettu 10.5.2015

Wikipedia: Ketterä ohjelmistokehitys.

http://fi.wikipedia.org/wiki/Ketterä_ohjelmistokehitys. Luettu 15.5.2015

Wikipedia: LDAP. <https://fi.wikipedia.org/wiki/LDAP>. Luettu 19.5.2015

Wikipedia: NetBeans. <http://en.wikipedia.org/wiki/NetBeans>. Luettu 21.5.2015

Wikipedia: Notepad++. <http://en.wikipedia.org/wiki/Notepad++>. Luettu 21.5.2015

Wikipedia: Ohjelmistotuotanto. <http://fi.wikipedia.org/wiki/Ohjelmistotuotanto>. Luettu 15.5.2015

Wikipedia: Projektinhallinta. <http://fi.wikipedia.org/wiki/Projektinhallinta>. Luettu 22.5.2015

Wikipedia: SQL. <https://fi.wikipedia.org/wiki/SQL>. Luettu 10.5.2015

Wikipedia: SQL-injektio. <https://fi.wikipedia.org/wiki/SQL-injektio>. Luettu 19.5.2015

Wikipedia: XAMPP. <http://en.wikipedia.org/wiki/XAMPP>. Luettu 21.5.2015